An Adaptive Method for Reducing Clock Skew in an Accumulative Z-Axis Interconnect System

Gary Bolotin
NASA: Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
gary.s.bolotin@jpl.nasa.gov

Lee Boyce
NASA: Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract This paper will present several methods for adjusting clock skew variations that occur in an accumulative z-axis interconnect system. In such a system, delay between modules is a function of their distance from one another. Clock distribution in a high-speed system, where clock skew must be kept to a minimum, becomes more challenging when module order is variable before design.

1. Introduction

The purpose of this paper is to inform you of a clock skew problem we encountered while implementing a PCI bus in a 3D stacked experimental flight computer for the X33 launch vehicle. This paper will first go into the background of the X-33 AFE project and will then present the clock skew problem in more detail by starting with an example of how this problem is solved in a backplane design. Finally, this paper will present how our design, a cumulative z-axis interconnect, is different than a backplane design and how we solved the clock skew problem.

The X-33 Launch Vehicle is a first step in the development of VentureStar. VentureStar is a planned Single-Stage to Orbit reusable launch vehicle that the companies listed below are designing. The VentureStar Launch Vehicle takes-off vertically, enters orbit, delivers its payload, returns to Earth and lands like an airplane - all on a single tank of gas. X33 is a prototype vehicle half the size of VentureStar or a 1/8 volume scale model, used to test out all of the new technologies required to make VentureStar work.

Since reducing weight is of great importance to the program, JPL was to design an experimental flight computer that would be 1/2 the size of the VME rack design currently used on X33. This product was to be produced first as experiment and not play a controlling role in the spacecraft. The experiment was called the X-33 Avionics Flight Experiment.(X-33 AFE)

2. X-33 Avionics Flight Experiment

For the X-33 AFE we proposed a 3D "stacked" PCI based flight computer, based on stacked PWB technology. The stacked module approach utilizes a vertical interconnect system, and thus eliminates the need for a backplane interconnect. This system has several advantages over a traditional backplane approach, the two most important are decreased overall mass and volume, and design flexibility. Since the modules interconnect directly to each other, the mass of a backplane is saved. In addition, unlike a backplane based systems, additional modules can be added without redesigning the interconnect system.

A similar but less dense technology of stacked PWBs would be a PC104 based system. Several systems have been built based on stacked MCMs[1,2,3]. This technology could be incorporated at a latter date if further volume reduction is necessary.

The X-33 AFE stack is comprised of stacking circuit slices, which have connector pads on two ends. These slices are interconnected with a an elastomeric connector from Amp Corporation. Figure 1 shows the stack in its assembled configuration. The system is made up by starting with the system computer slice, which holds the CPU, memory controller and PCI bridge. The rest of the system is constructed by other I/O slices were design as PCI devices or memory expansion on the CPU's local bus. The X-33 AFE consists of the following slices:

1. PowerPC 603 System Computer Slice
2. Main Memory Slice
3. Nonvolatile memory Slice
4. Two 3 Channel 1553 bus interface PCI slices
5. 1773 bus interface slice

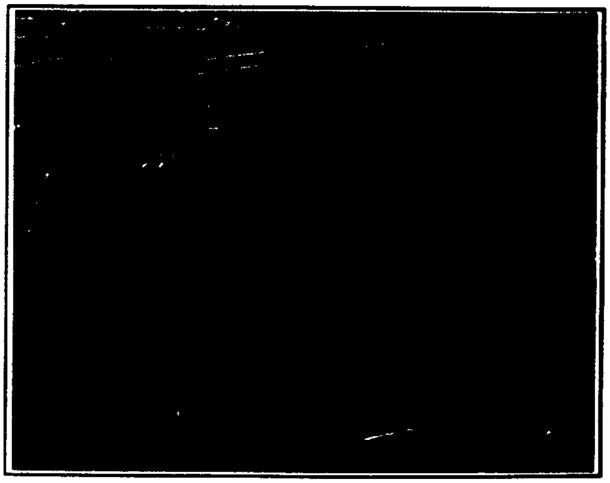

Figure 1. X-33 Avionics Flight Experiment

As shown in figure 2, pin layouts exist at each end of the board. Connections on one side implement a PCI bus, while connection on the other slice implement the processors local bus

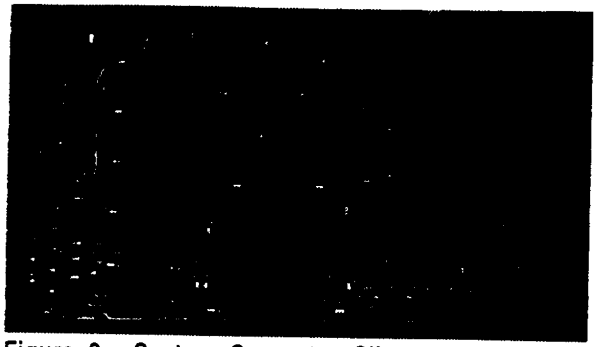

Figure 2. System Computer Slice

Figure 3 show the slice to slice interconnect mechanism. The connections are made by means of a vertical connector called an elastomer. These elastomers slip into fiberglass holders which are placed between the slices. The entire assembly is then bolted together with brackets at the ends to reduce any bowing, and provide compression.

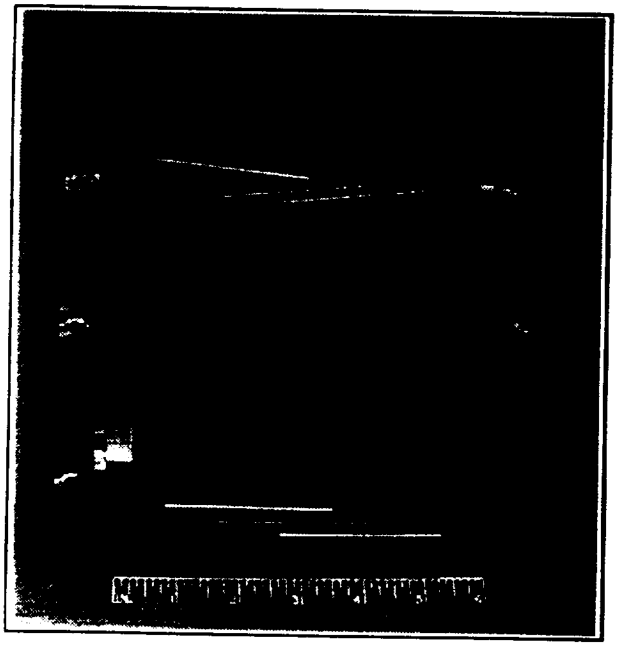

Figure 3. Interconnect System

1. The Clock Skew Problem

To understand the problem you need to understand a little about how PCI bus works.[4] PCI is a synchronous bus protocol. All transactions occur on the rising edge of the PCI clock that is received by all PCI devices. An Initiator is a device that starts a transaction and a Target is a device that responds. The top half figure 4 shows the beginning of a transaction where there is no clock skew between the Initiator and Target both devices see the identical clock timing.

In the first clock cycle the Initiator decides to start a transaction and asserts its FRAME# line about the same time it also places the Target's address on the bus. At the start of the second clock cycle the Target sees that FRAME# is asserted, and according to PCI requirements it latches what is on the address bus on the same clock edge. It will then take 1 to 3 clock cycles to decide whether the transaction is intended for it or not.

That's what is supposed to happen if all goes well. In the case of clock skew, the clock between an Initiator and a Target were skewed about 3 ns as shown in the bottom diagram. The clock of the initiator is shown in black above the target which is shown in red. This skew would have gone unnoticed if not for the fact that the Initiator, a MPC106 chip, was so fast. In the first clock cycle the Initiator would assert FRAME# within 2ns of the rising edge of the clock it saw. Since the Target's clock was skewed by more than this, plus the travel time of the signal, it saw frame asserted in clock cycle 1 instead of clock cycle 2 as in the previous case, and would latch whatever was on the address bus in that clock cycle. Though the Initiator asserted FRAME# quickly, it longer then the set-up time for the 32 lines of the address bus to stabilize , resulting in the Target latching in potentialy garbage data.

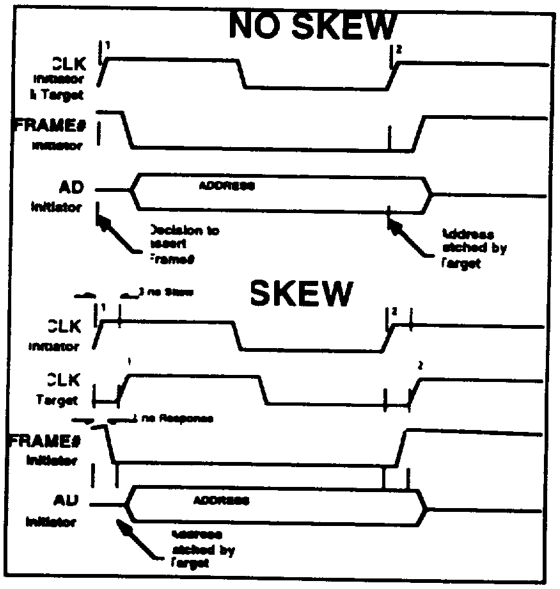

Figure 4. The Clock Skew Timing Problem

2. Compact PCI's Backplane Solution

In a 4 slot compact PCI backplane, the first slot is for the system card or main computer and clock generation, while the other three slots are for peripheral plug-in cards. Compact PCI (CPCI) solves this problem quite simply.[5] CPCI is design to ensure that the clock lines going to each peripheral card are all the same length, since the impedance of all the connectors are the same and the trace impedance on the backplane is constant, This results in the delay from the system master to any peripheral card a constant.

In addition, clock length is spec'ed for all PCI plug-in cards as the following diagram shows. Clock lengths are to be 2.5±0.1" while data lines, to give them a head start in a race condition are to be less than 1.5".

The total clock signal length is the sum of the distance from the clock generator on the system board, plus the distance through the 3U connector, length along the backplane back up through the peripheral board's connector and finally to the chip on the peripheral board.

CPCI tightly specifies the distance from an edge connector to the component on any peripheral board. Since these values are also fixed, CPCI design makes up for any differences in clock length by serpentining the clock lines through the backplane, as shown in the following figure. Clock distribution is therefore backplane dependent and left to the discretion of the designer. In this design both clock lines are equalized, since the distance traveled through the backplane is equal. To make the design easier, clocks are shared by T-ing the line between the two middle slots.

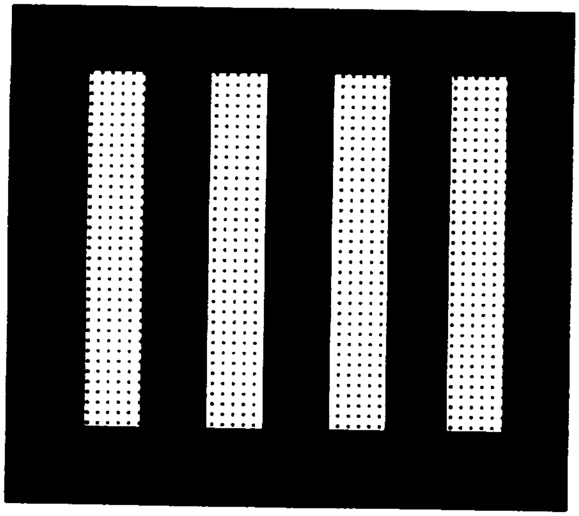

Figure 5. A 4 Slot Compact PCI Backplane

3. Z-Axis Interconnect

In a cumulative z-interconnect design, boards are stacked on top of the system slice, the bussed lines go through additional elastomeric connectors. The fact that the farther you are away from the master the more connectors you need to go through results in additional delay for some modules. In this picture, you can see that the clock line for PCI Device 1 is shorter than PCI Device 2. Since there is no backplane this leaves us with fewer options to equalize the clock lengths among the peripheral slices. .

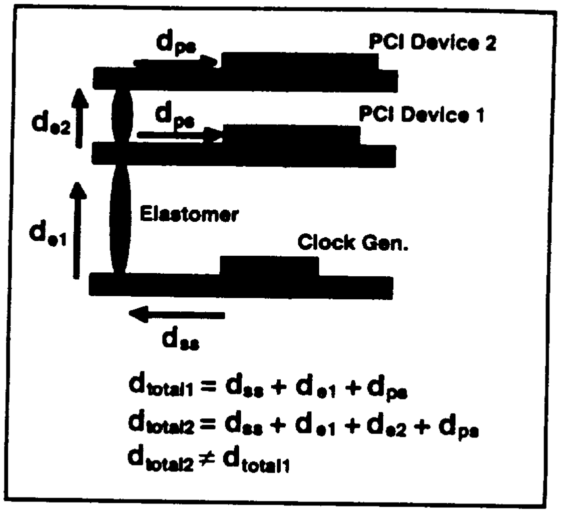
d_{pe}
PCI Device 2
d_{e2}
d_{pe}
PCI Device 1
d_{e1}
Elastomer
Clock Gen.
d_{ss}
$d_{total1} = d_{ss} + d_{e1} + d_{pe}$
$d_{total2} = d_{ss} + d_{e1} + d_{e2} + d_{pe}$
$d_{total2} \neq d_{total1}$

Figure 6. Z-Axis Cumulative Interconnect

For our design we chose the following design objectives. We wanted to eliminate clock skew, but we also desired the design of the slices to allow them to be positioned anywhere in the stack. We wanted to have some generic rules to apply to all of our PCI Peripheral Slices. We also wanted to use an adapter board to convert from our elastomeric connector to CPCI. This would allow us to use off the shelf CPCI bus analyzers and Ethernet card.

4. Z-Axis Interconnect Solution

Figure 7 shows one of the steps in how we solved the problem. By making all of the clock lines the same electrical length by varying delay lines lengths located on the system slice. I say electrical because as you go through each elastomer, you add additional capacitance to the clock line and round it over and thus adding an additional delay.

Let's look at the length of the clock line to the nearest peripheral slice, the one directly above the system board. From the clock generator, the signal goes through a delay line to the edge of the system board through the elastomer and down a given distance to the peripheral slices PCI component. For any other board some k-elastomers away from the system board, we shortened the delay line to compensate. The result is the sum of all of the lengths is more or less constant.

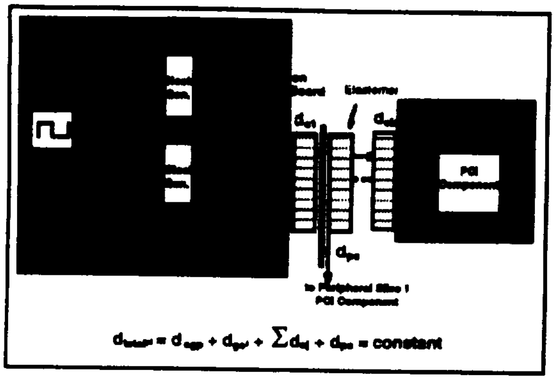

Figure 7. Clock Design Rules

In addition, we placed the system slice in the center of the stack and worked with the clocks symmetrically outward. This is illustrated in Figure 8. So two slices an equal distance away from the system slice share the same clock. We also pulled in the distance from the elastomeric pad layout to the PCI component so when the slice was attached to an adapter board, the assembly fell close to the 2.5" clock line requirement.

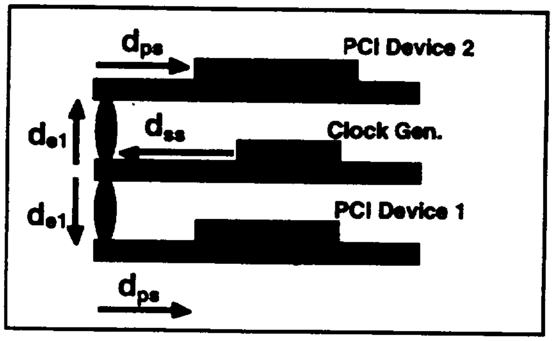

Figure 8. Additional Design Rules

5. Implementation - Conclusion

Figure 9 is a picture of the actual implementation. The picture is made uglier by our bringing a connector out by pigtails to the side of the board, for system debug purposes.

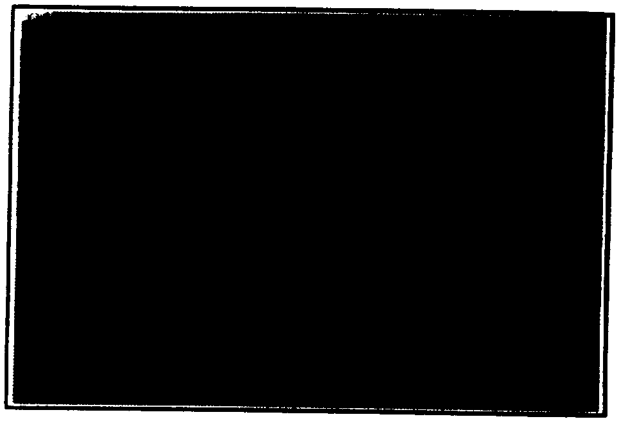

Figure 9 Implementation

You can clearly see the significant lines. The long one corresponding to the board closest to the system slice and the short jumper wire for the clock line to the slices farthest away. We used 28 gage coax to make the delay lines and only jumperéd the slices differently depending on their position.

We had a stack of 5 PCI devices and two memory boards, and we were able to attach an adapter board and use an external Ethernet card and a CPCI bus analyzer.

Although this is not the most elegant solution in the world, the most important thing is that it worked.

6. Acknowledgment

This research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration

7. References

[1] Minehan W. T, Weidner W. T.,. "Fabrication, Assembly, and Characterization of Stacked Multichip Modules Using Hot Pressed, C0-Fired Aluminum Nitride", International Conference on Multichip Modules (MCM 94), pp 356-361 20, April 1994.

[2] N. J. Teneketges, W. J. Jacobi and L. A. Wadsworth, "A High-Performance MCM-Based Spaceborne Processor", ICMCM Proceedings '92.

[3] Alkalaj L., Jarvis B.., "The Design and Implentation of NASA's Advanced Flight Computing Module," International Conference on Multichip Modules (MCM 93), pp 40-44, April1993.

[4] PICMG. "CompactPCI Specification, PICMG 2.0 R2.1", September 2, 1997.

5] PICMG. "PCI Bus Specification, PICMG 2.0 R2.1", September 2, 1997

An Adaptive Method for Reducing Clock Skew in an Accumulative Z-Axis Interconnect System

Gary Bolotin
NASA: Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
gary.s.bolotin@jpl.nasa.gov

Lee Boyce
NASA: Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract This paper will present several methods for adjusting clock skew variations that occur in an accumulative z-axis interconnect system. In such a system, delay between modules is a function of their distance from one another. Clock distribution in a high-speed system, where clock skew must be kept to a minimum, becomes more challenging when module order is variable before design.

1. Introduction

The purpose of this paper is to inform you of a clock skew problem we encountered while implementing a PCI bus in a 3D stacked experimental flight computer for the X33 launch vehicle. This paper will first go into the background of the X-33 AFE project and will then present the clock skew problem in more detail by starting with an example of how this problem is solved in a backplane design. Finally, this paper will present how our design, a cumulative z-axis interconnect, is different than a backplane design and how we solved the clock skew problem.

The X-33 Launch Vehicle is a first step in the development of VentureStar. VentureStar is a planned Single-Stage to Orbit reusable launch vehicle that the companies listed below are designing. The VentureStar Launch Vehicle takes-off vertically, enters orbit, delivers its payload, returns to Earth and lands like an airplane - all on a single tank of gas. X33 is a prototype vehicle half the size of VentureStar or a 1/8 volume scale model, used to test out all of the new technologies required to make VentureStar work.

Since reducing weight is of great importance to the program, JPL was to design an experimental flight computer that would be 1/2 the size of the VME rack design currently used on X33. This product was to be produced first as experiment and not play a controlling role in the spacecraft. The experiment was called the X-33 Avionics Flight Experiment.(X-33 AFE)

2. X-33 Avionics Flight Experiment

For the X-33 AFE we proposed a 3D "stacked" PCI based flight computer, based on stacked PWB technology. The stacked module approach utilizes a vertical interconnect system, and thus eliminates the need for a backplane interconnect. This system has several advantages over a traditional backplane approach, the two most important are decreased overall mass and volume, and design flexibility. Since the modules interconnect directly to each other, the mass of a backplane is saved. In addition, unlike a backplane based systems, additional modules can be added without redesigning the interconnect system.

A similar but less dense technology of stacked PWBs would be a PC104 based system. Several systems have been built based on stacked MCMs[1,2,3]. This technology could be incorporated at a latter date if further volume reduction is necessary.

The X-33 AFE stack is comprised of stacking circuit slices, which have connector pads on two ends. These slices are interconnected with a an elastomeric connector from Amp Corporation. Figure 1 shows the stack in its assembled configuration. The system is made up by starting with the system computer slice, which holds the CPU, memory controller and PCI bridge. The rest of the system is constructed by other I/O slices were design as PCI devices or memory expansion on the CPU's local bus. The X-33 AFE consists of the following slices:

1. PowerPC 603 System Computer Slice
2. Main Memory Slice
3. Nonvolatile memory Slice
4. Two 3 Channel 1553 bus interface PCI slices
5. 1773 bus interface slice

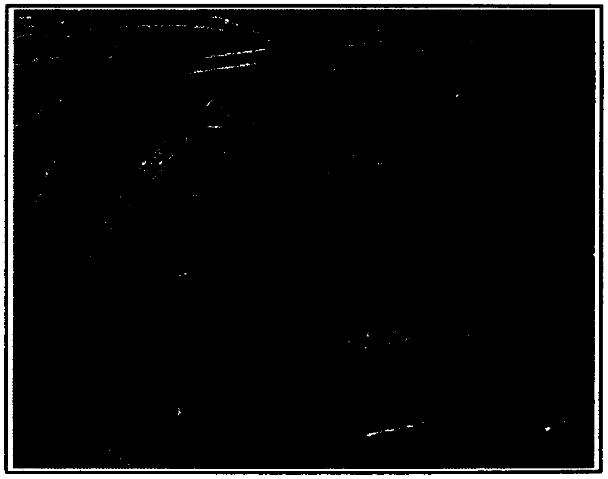

Figure 1. X-33 Avionics Flight Experiment

As shown in figure 2, pin layouts exist at each end of the board. Connections on one side implement a PCI bus, while connection on the other slice implement the processors local bus

Figure 2. System Computer Slice

Figure 3 show the slice to slice interconnect mechanism. The connections are made by means of a vertical connector called an elastomer. These elastomers slip into fiberglass holders which are placed between the slices. The entire assembly is then bolted together with brackets at the ends to reduce any bowing, and provide compression.

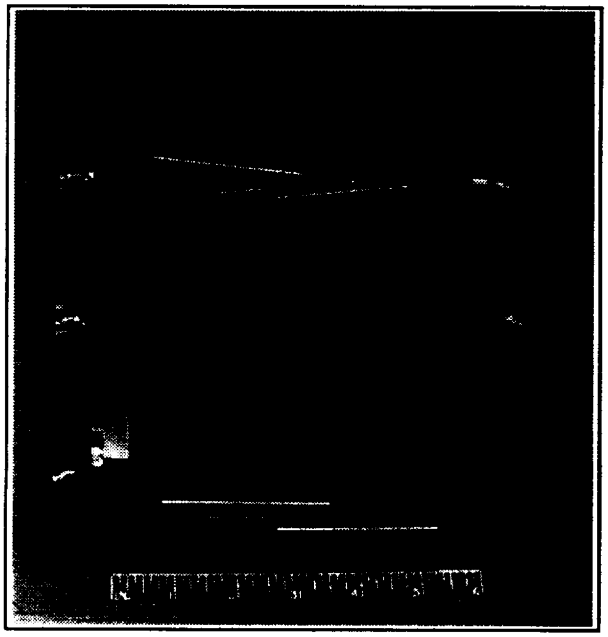

Figure 3. Interconnect System

1. The Clock Skew Problem

To understand the problem you need to understand a little about how PCI bus works.[4] PCI is a synchronous bus protocol. All transactions occur on the rising edge of the PCI clock that is received by all PCI devices. An Initiator is a device that starts a transaction and a Target is a device that responds. The top half figure 4 shows the beginning of a transaction where there is no clock skew between the Initiator and Target, both devices see the identical clock timing.

In the first clock cycle the Initiator decides to start a transaction and asserts its FRAME# line about the same time it also places the Target's address on the bus. At the start of the second clock cycle the Target sees that FRAME# is asserted, and according to PCI requirements it latches what is on the address bus on the same clock edge. It will then take 1 to 3 clock cycles to decide whether the transaction is intended for it or not.

That's what is supposed to happen if all goes well. In the case of clock skew, the clock between an Initiator and a Target were skewed about 3 ns as shown in the bottom diagram. The clock of the initiator is shown in black above the target which is shown in red. This skew would have gone unnoticed if not for the fact that the Initiator, a MPC106 chip, was so fast. In the first clock cycle the Initiator would assert FRAME# within 2ns of the rising edge of the clock it saw. Since the Target's clock was skewed by more than this, plus the travel time of the signal, it saw frame asserted in clock cycle 1 instead of clock cycle 2 as in the previous case, and would latch whatever was on the address bus in that clock cycle. Though the Initiator asserted FRAME# quickly, it longer then the set-up time for the 32 lines of the address bus to stabilize , resulting in the Target latching in potentialy garbage data.

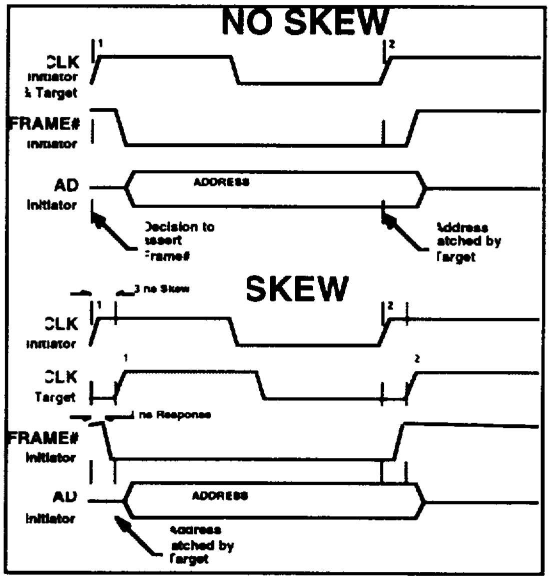

Figure 4. The Clock Skew Timing Problem

2. Compact PCI's Backplane Solution

In a 4 slot compact PCI backplane, the first slot is for the system card or main computer and clock generation, while the other three slots are for peripheral plug-in cards. Compact PCI (CPCI) solves this problem quite simply.[5] CPCI is design to ensure that the clock lines going to each peripheral card are all the same length, since the impedance of all the connectors are the same and the trace impedance on the backplane is constant, This results in the delay from the system master to any peripheral card a constant.

In addition, clock length is spec'ed for all PCI plug-in cards as the following diagram shows. Clock lengths are to be 2.5±0.1" while data lines, to give them a head start in a race condition are to be less than 1.5".

The total clock signal length is the sum of the distance from the clock generator on the system board, plus the distance through the 3U connector, length along the backplane back up through the peripheral board's connector and finally to the chip on the peripheral board.

CPCI tightly specifies the distance from an edge connector to the component on any peripheral board. Since these values are also fixed, CPCI design makes up for any differences in clock length by serpentining the clock lines through the backplane, as shown in the following figure. Clock distribution is therefore backplane dependent and left to the discretion of the designer. In this design both clock lines are equalized, since the distance traveled through the backplane is equal. To make the design easier, clocks are shared by T-ing the line between the two middle slots.

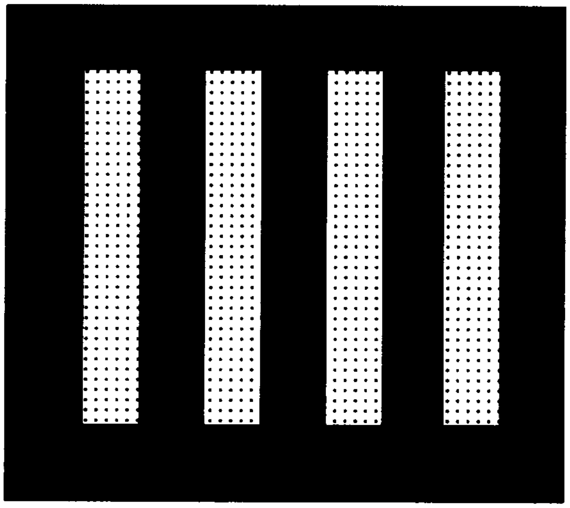

Figure 5. A 4 Slot Compact PCI Backplane

3. Z-Axis Interconnect

In a cumulative z-interconnect design, boards are stacked on top of the system slice, the bussed lines go through additional elastomeric connectors. The fact that the farther you are away from the master the more connectors you need to go through results in additional delay for some modules. In this picture, you can see that the clock line for PCI Device 1 is shorter than PCI Device 2. Since there is no backplane this leaves us with fewer options to equalize the clock lengths among the peripheral slices. .

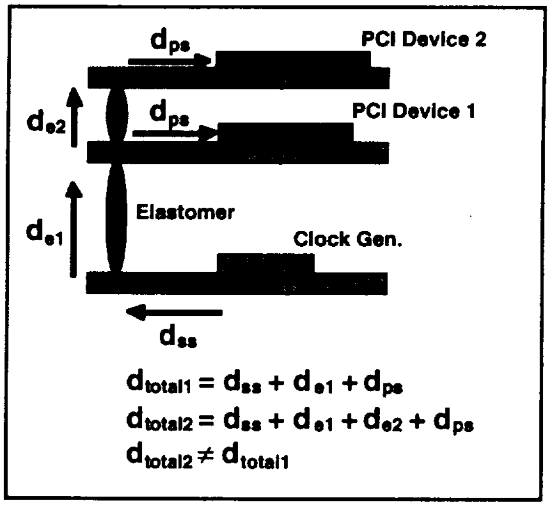
d_{ps}
PCI Device 2
d_{e2}
d_{ps}
PCI Device 1
d_{e1}
Elastomer
Clock Gen.
d_{ss}
$d_{total1} = d_{ss} + d_{e1} + d_{ps}$
$d_{total2} = d_{ss} + d_{e1} + d_{e2} + d_{ps}$
$d_{total2} \neq d_{total1}$

Figure 6. Z-Axis Cumulative Interconnect

For our design we chose the following design objectives. We wanted to eliminate clock skew, but we also desired the design of the slices to allow them to be positioned anywhere in the stack. We wanted to have some generic rules to apply to all of our PCI Peripheral Slices. We also wanted to use an adapter board to convert from our elastomeric connector to CPCI. This would allow us to use off the shelf CPCI bus analyzers and Ethernet card.

4. Z-Axis Interconnect Solution

Figure 7 shows one of the steps in how we solved the problem. By making all of the clock lines the same electrical length by varying delay lines lengths located on the system slice. I say electrical because as you go through each elastomer, you add additional capacitance to the clock line and round it over and thus adding an additional delay.

Let's look at the length of the clock line to the nearest peripheral slice, the one directly above the system board. From the clock generator, the signal goes through a delay line to the edge of the system board through the elastomer and down a given distance to the peripheral slices PCI component. For any other board some k-elastomers away from the system board, we shortened the delay line to compensate. The result is the sum of all of the lengths is more or less constant.

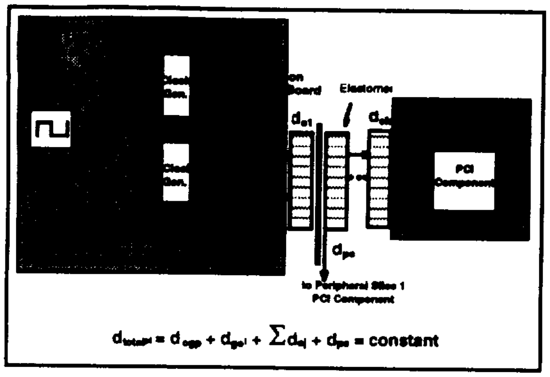
Clock Gen.
Clock Gen.
on Board
Elastomer
d_{e1}
d_{ps}
PCI Component
to Peripheral Slice 1 PCI Component
$d_{total} = d_{sgp} + d_{gel} + \sum d_{ej} + d_{ps}$ = constant

Figure 7. Clock Design Rules

In addition, we placed the system slice in the center of the stack and worked with the clocks symmetrically outward. This is illustrated in Figure 8. So two slices an equal distance away from the system slice share the same clock. We also pulled in the distance from the elastomeric pad layout to the PCI component so when the slice was attached to an adapter board, the assembly fell close to the 2.5" clock line requirement.

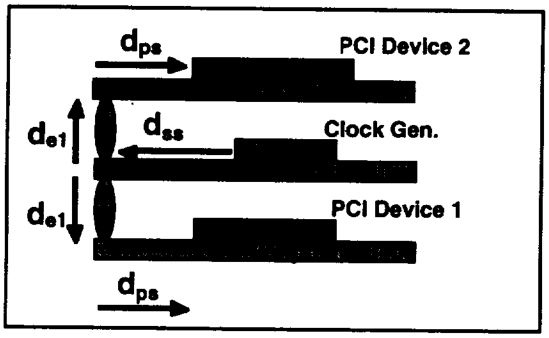

Figure 8. Additional Design Rules

5. Implementation - Conclusion

Figure 9 is a picture of the actual implementation. The picture is made uglier by our bringing a connector out by pigtails to the side of the board, for system debug purposes.

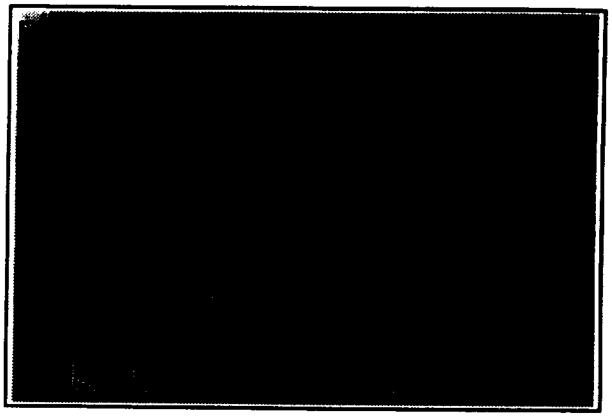

Figure 9 Implementation

You can clearly see the significant lines. The long one corresponding to the board closest to the system slice and the short jumper wire for the clock line to the slices farthest away. We used 28 gage coax to make the delay lines and only jumpered the slices differently depending on their position.

We had a stack of 5 PCI devices and two memory boards, and we were able to attach an adapter board and use an external Ethernet card and a CPCI bus analyzer.

Although this is not the most elegant solution in the world, the most important thing is that it worked.

6. Acknowledgment

This research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration

7. References

[1] Minehan W. T, Weidner W. T.,. "Fabrication, Assembly, and Characterization of Stacked Multichip Modules Using Hot Pressed, C0-Fired Aluminum Nitride", International Conference on Multichip Modules (MCM 94), pp 356-361 20, April 1994.

[2] N. J. Teneketges, W. J. Jacobi and L. A. Wadsworth, "A High-Performance MCM-Based Spaceborne Processor", ICMCM Proceedings '92.

[3] Alkalaj L., Jarvis B.., "The Design and Implentation of NASA's Advanced Flight Computing Module," International Conference on Multichip Modules (MCM 93), pp 40-44, April1993.

[4] PICMG. "CompactPCI Specification, PICMG 2.0 R2.1", September 2, 1997.

5] PICMG. "PCI Bus Specification, PICMG 2.0 R2.1", September 2, 1997